**ReadMe:**

**מגישות:**

רותם שמחוני- 205667892

ספיר חמצאני- 205986573

אימייל-   [sapirham@gmail.com](mailto:sapirham@gmail.com)

**הסבר כללי על מהלך המשחק ומבנה התוכנית:**

מבנה התוכנית:

* שלושה אובייקטים:
* שחקן שחור
* שחקן לבן
* משחק
* אובייקט שחקן מכיל את הנתונים הנדרשים לחישוב הסטטיסטיקות עבור כל שחקן ואת שם השחקן.
* אובייקט המשחק מכיל את נתוני הלוח , נתוני זמנים לחישוב ממוצעים, סה״כ התורות ושחקן נוכחי.

תחילת משחק:

* השחקנים מכניסים את שמותיהם, בוחרים גודל לוח ולוחצים על הכפתור let’s start  .
* לאחר מכן יועברו לדף התחלת משחק.

לפני כל מהלך:

* ישנה בדיקה האם השחקן שאמור לשחק יכול לבצע מהלך בלוח – כלומר יכול לקבל ניקוד עבור לחיצה מסוימת, אם כן התור יעבור לאותו שחקן אחרת יישאר על השחקן הנוכחי.
* הפונקציה שעושה בדיקה זו היא  chekAvailableMoves(). במידה וכן אפשר לבצע מהלכים , כל תא בטבלה שמייצג מהלך יישמר במערך  reverse.saveArrayToCgange שיכיל גם את אינדקסים התאים ש״ייאכלו״ בעקבות אותו מהלך.
* במידה ובשלב מסוים השחקן מעוניין לראות את פוטנציאל המהלכים עבור כל לחיצה על הכפתור, התוכנית משתמשת במערך זה כדי להציג את הניקוד הרלוונטי.
* בנוסף, פונקציית הבדיקה תדאג בסיומה ל**מנוע כל לחיצה** על תא שאינו מהווה מהלך עבור אותו שחקן ואף תצבע תא זה בצבע כהה, כף שבסופו של דבר לשחקן תהיה אפשרות לחיצה רק עבור התאים הבהירים והלחיצים!

ביצוע מהלך:

* שחקן יוכל לשים דיסקית בצבע שלו על הלוח רק במקומות בהם הוא מקבל ניקוד, לא ניתנת האופציה לשים דיסקית במקום בו השחקן אינו מרוויח כלל מהמהלך.
* לאחר שהשחקן ביצע מהלך על ידי לחיצה על תא (כפי שהסברנו לעיל) ישתנה הלוח בהתאם למהלך ובנוסף יעודכנו הסטטיסטיקות של שני המשתתפים.
* לפני העברת התור לשחקן הבא נבצע בדיקת מהלכים עבור השחקן הנגדי בעזרת הפונקציה checkAvailableMoves().
* במידה והשחקן הנגדי אינו יכול לשחק נשאיר את התור אצל השחקן הנוכחי.

סיום משחק:

* ישנם שלוש דרכים בהם ניתן לסיים משחק:
* הלוח מלא.
* במידה ושני השחקנים אינם יכולים לבצע מהלך.
* במידה ושחקן החליט ללחוץ על כפתור סיום משחק.

במקרים א ו ב השחקן המנצח ייקבע על פי סך הניקוד (כלומר סך הדיסקים בצבעים שלו שקיימים בלוח).

במקרה ג השחקן היריב (שאינו לחץ) מנצח ניצחון טכני ללא קשר למצב הניקוד.

* בכל שלב סיום תוצג כותרת בראש העמוד עם שם המנצח והשעון ייעצר כך שניתן יהיה לראות את הסטטיסטיקות הסופיות.

שעון המשחק:

* ישנם שני שעונים שרצים במהלך משחק:
* האחד נראה לעיני השחקנים ומראה כמה זמן עבר מתחילת המשחק.
* השני אינו נראה לעיניי השחקנים ומאותחל בכל תור מחדש- בעזרתו מחשבים את הממוצע למהלך עבור כל שחקן.

**בונוסים שמומשו:**

* קביעת גודל הלוח לפי בחירת המשתמש.
* בתחילת כל משחק בעת הכנסת שמות השחקנים ניתנת האפשרות לבחור את גודל הלוח.
* אופציה למשחק חדש עבור אותם שחקנים על ידי לחיצה על הכפתור Start Over:
* כפתור זה יינתן ללחיצה רק לאחר סיום משחק (עבור אחד משלושת המקרים שהוצגו).
* בנוסף הלוח יאותחל והסטטיסטיקות יעודכנו למצב ההתחלתי.
* כעת לאחר ביצוע מהלך יוצג לשחק זמן ממוצע לביצוע מהלך עבור המשחק החדש ובנוסף בinput נפרד זמן ממוצע לביצוע מהלך עבור כל המשחקים.
* אפשרות לראות בלחיצת כפתור את פוטנציאל המהלכים:
* על ידי לחיצה על הכפתור  Show potential יוצג לשחקן על הלוח ניקוד עבור כל מהלך שייעשה (במקומות בהם הלחיצה אפשרית).
* לחיצה על  Hide potential תסיר מהלוח את הניקוד עבור כל לחיצה.

**הסבר על פונקציות עיקריות:**

**checkIfRivalAround()**

* עבור כל תא בלוח היא בודקת האם קיים באחד מתשעת המקומות מסביב לתא זה שחקן יריב.
* עבור כל שחקן יריב שנמצא, אותו מקום (אחד מהתשעה) יישלח לפונקציה checkIfMoveSucced() -תוסבר בהמשך.
* במידה ולא נמצא שחקן יריב מסביב לתא הנבדק, תא זה יוכהה ולא יהיה ניתן ללחיצה.
* במידה וכן נמצא שחקן יריב אך עדיין לא ניתן לבצע מהלך, תא זה גם כן יוכהה ולא יהיה ניתן ללחיצה.

**checkIfMoveSucced()**

* פונקציה זו מקבלת תא ספציפי של השחקן היריב (אחד מהתשעה שנבדקו בפונקציה לעיל).
* הפונקציה מבצעת בדיקה האם אכן ניתן לבצע מהלך, כלומר האם בלחיצה על תא זה השחק יקבל ניקוד על ידי ״אכילה״ של השחקן היריב.
* במידה וכן, את כל התאים שייעודו להחלפה בצבע יישמרו במערך.
* מערך זה יוחזר חזרה לפונקציה checkIfRivalAround() ויוכנס כitem לתוך מערך saveArrayToChange
* לכל תא בלוח יש ייצוג אחד בלבד במערך זה כאובייקט , כך שאחד הפרמטרים זה מערך שכולל את כל האינדקסים שיוחלף צבעם בעקבות הלחיצה על התא (כלומר מערך שכולל את כל המערכים שחזרו עבור תא זה מהפונקציה .checkIfMoveSucceed()

**move()**

* פונקציה זו תקרא לאחר שהשחקן לחץ על אחד מתאי הלחיצה האפשריים.
* כלומר, בהתאם להסברים קודמים עבור תא זה בהכרח ניתן לבצע מהלך ולכן נחפש את התא במערך saveArrayToChang ונשנה את הצבע עבור כל אינדקס (כל תא) לצבע השחקן הנוכחי.
* כעת, נעדכן את נתוני הסטטיסטיקות של שני השחקנים בפונקציה  recalculateStatistics()
* בנוסף, פונקציה זו תבצע מספר בדיקות:
* האם הלוח מלא- כתוצאה מכך ייגמר המשחק.
* האם השחקן היריב יכול לשחק:
* אם כן, התור יועבר,
* אחרת בדיקה האם השחקן הנוכחי יכול לבצע מהלך:
* אם כן, השחקן הנוכחי יוכל לבצע מהלך נוסף.
* אחרת נגמר המשחק.

**recalculateStatistics()**

* מעדכן את הניקוד של שני השחקנים (הוספה והורדה בהתאם).
* מחשבת זמן ממוצע לביצוע מהלך על ידי חישוב סה”כ כל הזמן שלקח לבצע מהלכים לחלק לסה”כ התורות.
* במידה וזהו משחק שני ומעלה של שני השחקנים הנוכחים , פונקציה זו גם תחשב זמן לביצוע מהלך ממוצע עבור כל המשחקים על ידי חישוב סה”כ כל הזמן שלקח לבצע מהלכים (במשחק הנוכחי ובמשחקים האחרים) לחלק לסה”כ התורות (במשחק הנוכחי ובמשחקים האחרים).
* כעת, נעדכן את הסטטיסטיקות בדף המשחק בפונקציה updateInputs()

**initGame()**

* פונקציית אתחול שנקראת לאחר טעינת דף HTML  בפונקציה זו נבצע:
* קריאה לפונקציית  onClick()   עבור כל הכפתורים במשחק.
* קריאה לפונקציות אתחול נוספות.

**preparGrid()**

* בפונקציה זו נבנה את לוח המשחק על פי הגודל הנבחר על ידי השחקנים.
* תתבצע קריאה לפונקציית  onClick()  של כל אחד מהתאים.
* יאותחלו ארבעת התאים המרכזיים ב-2 עיגולים שחורים ו-2עיגולים לבנים.

**addTick()**

* פונקציה שמדמה שעון ונקראת בכל שנייה.
* מעדכנת את הזמן שעבר מתחילת המשחק ואת הזמן שעבר עבור כל תור ספציפי בשניות ובדקות.